Global Impostor Selection for DBNs in Multi-session i-Vector Speaker Recognition

Omid Ghahabi and Javier Hernando

TALP Research Center, Department of Signal Theory and Communications
Universitat Politecnica de Catalunya - BarcelonaTech, Spain
{omid.ghahabi,javier.hernando}@upc.edu

Abstract. An effective global impostor selection method is proposed in this paper for discriminative Deep Belief Networks (DBN) in the context of a multi-session i-vector based speaker recognition. The proposed method is an iterative process in which in each iteration the whole impostor i-vector dataset is divided randomly into two subsets. The impostors in one subset which are closer to each impostor in another subset are selected and impostor frequencies are computed. At the end, those impostors with higher frequencies will be the global selected ones. They are then clustered and the centroids are considered as the final impostors for the DBN speaker models. The advantage of the proposed method is that in contrary to other similar approaches, only the background i-vector dataset is employed. The experimental results are performed on the NIST 2014 i-vector challenge dataset and it is shown that the proposed selection method improves the performance of the DBN-based system in terms of minDCF by 7% and the whole system outperforms the baseline in the challenge by more than 22% relative improvement.

Index Terms: Speaker Recognition, Deep Belief Network, Impostor Selection, NIST i-vector challenge.

1 Introduction

Speaker recognition based on identity vectors (i-vector) [1] is widely accepted as the state-of-the-art in this field. To compensate undesired speaker and session variabilities, i-vectors are further post-processed with some techniques [1][2][3][4]. A common and successful paradigm for multi-session speaker recognition is based on i-vector and Probabilistic Linear Discriminant Analysis (PLDA) in which a combination of different sessions will be carried out either in the i-vector or score level [5][6] [7]. In the i-vector level, the available i-vectors per each target speaker are averaged and the resulting i-vector is compared with the test i-vectors. On the other hand, in the score level combination, each i-vector belonging to the given target speaker is compared separately with the test i-vectors and then obtained scores are combined.

To encourage research groups to deal with new challenges, the National Institute of Standard and Technologies (NIST) organizes from time to time speaker

J.L. Navarro Mesa et al. (Eds.): IberSPEECH 2014, LNAI 8854, pp. 89–98, 2014.

recognition evaluations and the participating sites present their results in the Speaker Odyssey Workshop. The most recent challenge is planned for modeling i-vectors in a multi-session enrollment task [8]. A large amount of unlabeled i-vectors are given as a development set and participating sites are not allowed to use their own data set to develop the systems. Thus the supervised variability compensation techniques like Within-Class Covariance Normalization (WCCN), Linear Discriminant Analysis (LDA), or the most effective one (PLDA) cannot be used easily in this challenge as they need speaker labels. Moreover, i-vectors are extracted from speech utterances with different durations which it makes the challenge more difficult.

Most of participating sites tried to develope some automatic speaker classifiers on the development i-vectors to label data and then to use supervised learning approaches [9] [10][11][12]. Some of them also tried to solve the unmatch condition between train and test i-vectors from the speech duration point of view [11][10] . We proposed a Deep Belief Network (DBN) based system [13] in which we could achieve notable results in comparison to the baseline and other individual systems [9][10] [11][12] without using the speech duration information or any automatic labeling technique. In this paper we propose a new global impostor selection method which achieves similar results as in [13], in the context of our DBN-based system, by using only the background dataset. Using just the background dataset is actually the advantage of the proposed method in comparison to other similar ones [14][15][16][13], as they use also the training dataset as a part of the selection method.

In our DBN-based system we take the advantage of unsupervised learning to model a global DBN to be used in an adaptation process and the advantage of supervised learning to model each target speaker discriminatively. As more than one i-vector sample are available per each target speaker in this case and each of them may be recorded from different session, DBNs will capture more speaker and session variabilities from the input data and will work better than in the single session task [16].

2 i-Vector Extraction

This section gives a brief overview on the i-vector framework developed in [1]. Given the centralized Baum-Welch statistics from all available speech utterances, the low rank total variability matrix ($\mathbf{T}$) is trained in an iterative process. The training process assumes that an utterance can be represented by the Gaussian Mixture Model (GMM) mean supervector,

$$\mathbf{m} = \mathbf{m}_u + \mathbf{T}\boldsymbol{\omega} \tag{1}$$

where $\mathbf{m}_u$ is the speaker- and session-independent mean supervector from the Universal Background Model (UBM), and $\boldsymbol{\omega}$ is a low rank vector referred to as the identity vector or i-vector. The supervector $\mathbf{m}$ is assumed to be normally distributed with the mean $\mathbf{m}_u$ and the covariance $\mathbf{T}\mathbf{T}^t$, and the i-vectors have a standard normal distribution $\mathcal{N}(0, 1)$. More details can be found in [1].

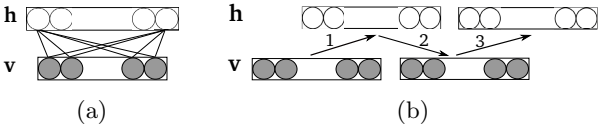


Fig. 1. RBM (a) and RBM training (b)

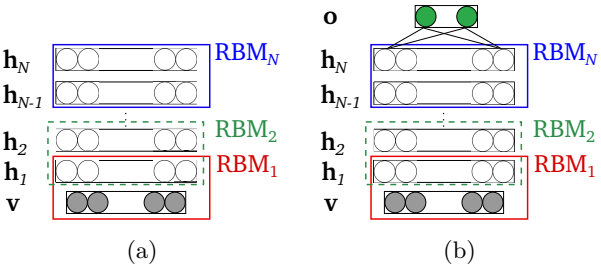


Fig. 2. Generative (a) and discriminative (b) DBNs

3 Deep Belief Networks

DBNs are originally probabilistic generative networks with multiple layers of stochastic hidden units above a layer of visible variables. There is an efficient greedy layer-wise algorithm for learning DBNs [17]. The algorithm treats every two adjacent layers as an RBM (Figs. 1a and 2a). The output of each RBM is considered as the input to its above RBM. RBMs are constructed from a layer of binary stochastic hidden units and a layer of stochastic visible units (Fig. 1a).

Training an RBM is based on an approximated version of the Contrastive Divergence (CD) algorithm [18][17] which consists of three steps (Fig. 1b). At first, hidden states (**h**) are computed given visible states (**v**), then given **h**, **v** is reconstructed, and in the third step **h** is updated given the reconstructed **v**. Finally, the change of connection weights is given as follows,

$$\Delta w_{ij} \approx -\alpha \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \right) \tag{2}$$

where α is the learning rate, w_{ij} represents the weight between the visible unit i and the hidden unit j, $\langle . \rangle_{data}$ and $\langle . \rangle_{recon}$ denote the expectations when the hidden state values are driven respectively from the input visible data and the reconstructed data. Actually, the training process tries to minimize the reconstruction error between the actual input data and the reconstructed one. The parameter updating process is iterated until the algorithm converges. Each iteration is called an epoch. It is possible to perform the above parameter update after processing each training example, but it is often more efficient to divide the whole input data (batch) into smaller size batches (minibatch) and to do the parameter update by an average over each minibatch. More theoretical and practical details can be found in [18][17][19].

When the unsupervised learning is finished, by adding a label layer on top of the network and doing a supervised backpropagation training, it can be converted to a discriminative model (Fig. 2b). In other words, the unsupervised

learning can be considered as a pre-training for the supervised stage. It has been shown [17] that this unsupervised pre-training can set the weights of the network to be closer to a good solution than random initialization and, therefore, avoids local minima when using supervised gradient descent.

4 i-Vector Modeling Using DBN

The main idea is to model discriminatively the target and impostor i-vectors by a DBN structure. The structure which was proposed for the first time in a single session enrollment task [16] and later in a multi-session one [13] will be used also in this paper. In this section, we describe briefly the whole structure used, and in the next section focus on the proposed impostor selection method which is the main new contribution of this paper. As illustrated in Fig. 3, the DBN structure is composed of three main parts namely balanced training, adaptation, and fine-tuning.

Like other discriminative methods, DBNs need also balanced positive and negative input data to achieve their best results. The balanced training part in the block diagram (Fig. 3) tries to use the information of all available impostors and decrease their population in a reasonable way. The decreasing is carried out in two steps, selecting the most informative ones and clustering. In [16] and [13] simple and effective selection methods are proposed. First, the n closest impostors to each target speaker are chosen according to their cosine distances. Then the closest impostors are accumulated over all target speakers and the k top ranked impostors are selected according to the number of times they are appeared in the accumulated set of impostors. In other words, the k impostors which are statistically closer to all target speakers are selected by this method. The selected impostors are clustered finally by the k-means algorithm using the cosine distance criterion.

In the multi-session task where more than one positive sample are available per each target speaker, we will choose the number of impostor cluster centroids in each minibatch the same as the number of available positive samples to make the training balanced. Hence, if the number of minibatches is set to three, for instance, and the number of positive samples per each speaker is five, the total number of impostor clusters will be 15. Actually, in each minibatch we will show the network the same positive samples as in other minibatches but different negative ones.

DBNs have the ability to be trained unsupervisingly [17][18] contrary to conventional neural networks that need labeled data to be trained. Thus a global model called Universal DBN (UDBN) [16] is trained by feeding many i-vectors from development background data. The training is carried out layer by layer using RBMs as described in section 3. UDBN parameters are adapted to the new data of each speaker including both target and impostor samples obtained in the balanced training part of Fig. 3. The adaptation is carried out by pre-training each network initialized by the UDBN parameters. It is shown [16] that the adaptation process outperforms both random and pre-training initializations.

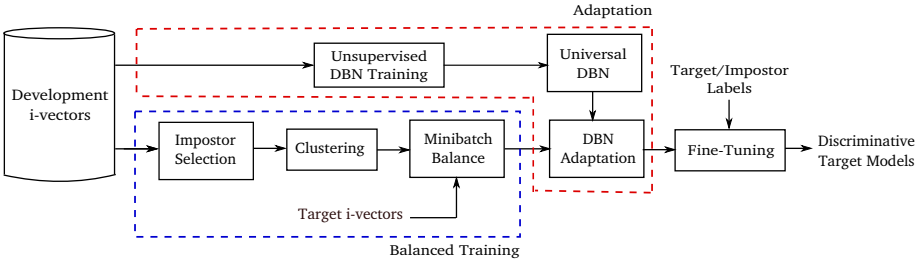


Fig. 3. Block-diagram of the DBN based speaker recognition system

Once the adaptation process is completed, a label layer is added on the top of the network and the stochastic gradient descent backpropagation is carried out as the fine-tuning process. The softmax will be the activation function of the top label layer. To minimize the negative effect of using random numbers used for initializing the top layer parameters, a pseudo pre-training process is performed by only one layer error backpropagating for a few iterations before a full backpropagation is carried out. If the input labels in the training phase are chosen as $(l_1 = 1, l_2 = 0)$ and $(l_1 = 0, l_2 = 1)$ for target and impostor i-vectors respectively, the final output score in the testing phase will be computed in a Log Likelihood Ratio (LLR) form as follows,

$$LLR = log(o_1) - log(o_2) \tag{3}$$

where (o_1, o_2) represents the outputs of the top layer. LLR computation helps to gaussianize the true and false score distributions.

5 Global Impostor Selection

As it was mentioned in section 4, we need to decrease the number of impostors in an effective way to provide a balanced training for DBNs. In [16] and [13] we decreased the number of impostors in two steps, impostor selection and clustering. In this paper we will focus on the impostor selection step. A support vector based method was proposed in [14] to select the most informative impostors for the Support Vector Machine (SVM) speaker recognition. In that work, those impostors which are selected more frequently as the support vectors in the target SVM models are shown to be more informative. In [15] the authors are extended the method proposed in [14] by pooling the target dependent support vectors and the global selected ones. In [16] and [13] we proposed a method in which those impostors statistically close to all the target i-vectors are selected as the final impostors. We used cosine distance criterion as a similarity measure in our method. However, in all of these methods the target speakers play a main role in the selection process.

The idea in this paper is to develop an impostor selection method which uses only the development background data keeping the advantage of being global to all the target speakers. The base of the proposed method will be the same as

in [16] but we show that by substituting the target i-vectors with a randomly selected subset of the background dataset we can keep the performance of the selection method.

In the proposed algorithm, the whole background dataset B is divided randomly into two subsets B_1 and B_2 where $B_1 \cup B_2 = B$, $B_1 \cap B_2 = \phi$, $|B_1| \ll |B_2|$, and $|.|$ denotes the number of members in a set. Then each i-vector in subset B_1 is compared with all i-vectors in subset B_2 using cosine distance criterion. The n i-vectors in B_2 which are closest to each i-vector in B_1 are selected. The number of times that each i-vector in B_2 is selected will be referred to as impostor frequency and is considered as a measure for the importance of that impostor in the whole background dataset B. To make the process statistically more reliable, we repeat the whole process several times (100 times in our experiments) and in each iteration we accumulate the impostor frequencies. We assume that higher impostor frequencies correspond to higher importance of those impostors in comparison to other ones in the background data. Thus we sort the impostor frequencies descendingly at the end and select the first k ones as the final impostors. It is worth noting that since the diagram of the sorted frequencies will be smooth enough due to the repetition of the selection process, we just select the first k impostors with the higher frequencies. In this way, there will be no need to use a threshold for selecting the impostors as it is proposed in [13].

The whole selection procedure can be summarized as follows.

1. Initialization
   (a) $t = 1$
   (b) $f_m = 0, 1 \leq m \leq M$
2. Divide the background i-vector dataset $B = \{\boldsymbol{\omega}_m, 1 \leq m \leq M\}$ into B_1 and B_2, where $B_1 = \{\boldsymbol{\nu}_i, 1 \leq i \leq 1000\}$, $B_2 = \{\boldsymbol{\chi}_j, 1 \leq j \leq M - 1000\}$, $B_1 \cup B_2 = B$, and $B_1 \cap B_2 = \phi$
3. For each $\boldsymbol{\nu}_i \in B_1$
   (a) Compute $score\left(\boldsymbol{\nu}_i, \boldsymbol{\chi}_j\right), 1 \leq j \leq M - 1000$
   (b) Select the n i-vectors $\boldsymbol{\chi}_j$ with the highest scores
   (c) Search for the corresponding indexes m of the selected i-vectors
   (d) For the selected i-vectors $f_m \leftarrow f_m + 1$
4. $t \leftarrow t + 1$
5. if $t \leq 100$ go to 2
6. Sort $f_m, 1 \leq m \leq M$ descendingly
7. Select the first k i-vectors as the final impostors,

where $score\left(\boldsymbol{\nu}_i, \boldsymbol{\chi}_j\right)$ is the cosine score between two i-vectors $\boldsymbol{\nu}_i$ in set B_1 and $\boldsymbol{\chi}_j$ in set B_2. The values of 1,000 and 100 are set arbitrary for the size of B_1 and the number of iterations, respectively. The parameters n and k will be determined experimentally in section 6.

6 Multi-session Experiments

The details of the database, the baseline and the DBN-based setups, and the obtained results are given in this section.

6.1 Baseline and Database

The experiments are carried out on the NIST 2014 i-vector challenge [8]. In this challenge contrary to other previous NIST evaluations, i-vectors are provided instead of speech signals. The i-vectors are computed from conventional telephone speech recordings in the SRE 2004 to 2012. The durations of speech utterances used to obtain i-vectors are different. They are sampled from a normal distribution with a mean of 40 s. The length of each i-vector is 600. Three sets of i-vectors are provided: unlabeled development, model, and test. The amounts of i-vectors in each set are respectively 36,572, 6,530, and 9,634. The number of target models is 1,306 and for each of them five i-vectors are available. Each model will be scored against all the test i-vectors and, therefore, 12,582,004 trials will be reported. Among all trials, 40% (progress subset) will be scored by NIST as a feedback to develop the system and 60% (evaluation subset) will be reserved for the final official evaluation. The performance is evaluated using a new Decision Cost Function (DCF) defined by NIST [8],

$$DCF(t) = (\#Miss(t)/\#Targets) + 100 \times (\#FalseAlarm(t)/\#NonTargets) \quad (4)$$

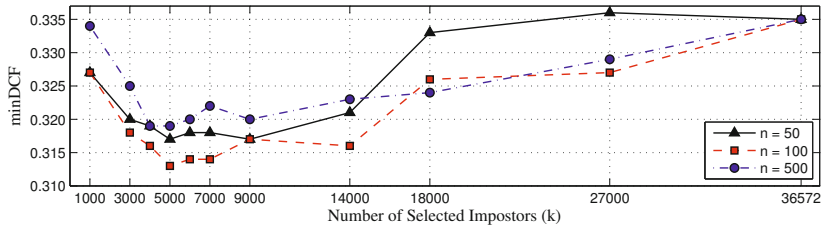
where t refers to the threshold for which the DCF is being computed. The minimum DCF obtained over all thresholds will be the official system score.

In the baseline system, average i-vectors obtained over the available i-vectors for each target speaker are scored against all test i-vectors using cosine distance classifier. However, before averaging and scoring some post-processing is carried out on i-vectors. The global mean and covariance are computed using unlabeled development data. All i-vectors are centered and whitened based on the global mean and covariance. Then the resulting i-vectors are length normalized. Length normalization is applied again on the average i-vectors obtained for each target speaker.

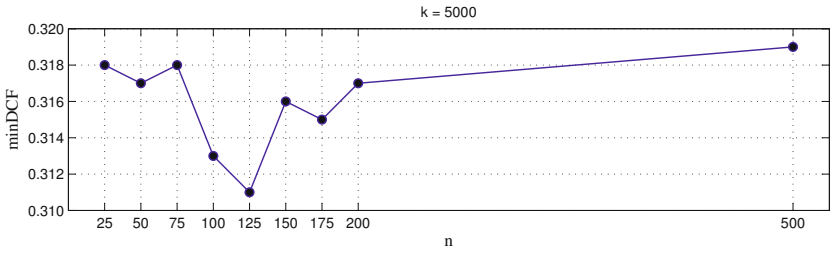
6.2 DBN-Based Setup

As in [16] DBNs with only one hidden layer are explored in this paper. The size of hidden layer is set to 400. Each minibatch will include five impostor centroids and five target samples. The impostor centroids in each minibatch are different than those in other ones, but they share the same target samples. The number of minibatches is set to three and, therefore, we will have 15 impostor centroids in total. The unlabeled development i-vectors provided by NIST are used for impostor selection. UDBN is trained with the same development i-vectors as in the impostor database. As the input i-vectors are real-valued normal distributed, a Gaussian-Bernoulli RBM [19][20] is employed. The learning rate (α), the number of epochs (NofE), and the minibatch size are set respectively to 0.02, 50, and 100 for UDBN training. A fixed momentum of 0.9 and a weight decay of 2×10^{-4} are also considered.

The adaptation process is carried out with $\alpha = 0.03$ and NofE=25. To decrease the probability of overfitting during the adaptation, it is performed on each minibatch separately and then the obtained network parameters are averaged.



(a)



(b)

Fig. 4. Determination of the parameters n and k for the proposed global imposter selection method (a) finding the rough values (b) keeping parameter k fixed and seting parameter n

The softmax connection weights are initialized by $\mathcal{N}(0, 0.01)$ and pre-trained with $\alpha = 1$ and NofE=15 before the whole backpropagation is performed. The momentum is started by 0.4 and is scaled up by 0.1 after each epoch (up to 0.9). The whole backpropagation is then carried out with $\alpha = 1$, NofE=30, and a fixed momentum of 0.9. The weight decay for both top layer pre-tarining and the whole backpropagation is set to 0.0014.

6.3 Results

Figure 4a illustrates the variability of minDCF obtained by eq. 4 in terms of the two parameters n and k defined in sec. 5. We have selected three rough values for parameter n and have plotted the obtained minDCF values for different number of selected global impostors. The figure shows that for all three selected values for parameter n, the minimum value of minDCF is obtained by selecting the first 5,000 impostors with higher frequencies ($k = 5,000$). Moreover, the figure shows that a medium value of $n = 100$ archives better results in overall. Therefore, $n = 100$ and $k = 5,000$ are chosen as the rough values which the selection method can perform well. By keeping one parameter fixed and setting another one and vice versa we can set both parameters more accurately. Figure 4b illustrates the variation of minDCF in terms of parameter n when $k = 5,000$. It shows that $n = 125$ yields the best result when k is kept fixed. Our experimental results

Table 1. Performance comparison of the DBN-based system with the baseline. The results are obtained on the NIST 2014 i-vector challenge.

System	Impostors	minDCF
Baseline	-	0.386
DBN-based	Full Background	0.335
DBN-based	Global Selected	0.311
DBN-based	Global Selected + 200 closest Target-Dependent	0.300

show that keeping $n = 125$ and changing k does not achieve better results. Thus we set $n = 125$ and $k = 5,000$ for our impostor selection method.

Table 1 shows the importance of the proposed impostor selection method in comparison to when the whole background dataset is used before clustering in fig. 3. As it can be seen in this table the global impostor selection method helps the DBN system to be around 7% more efficient. The experimental results show that if we pool the global selected impostors with 200 closest background i-vectors to each target speaker and then cluster them all together for each target speaker independently, we will achieve better results (minDCF=0.300) although it would be more computationally expensive. The overall performance of our DBN-based system is also notable (22% relative improvement) in comparison to the baseline in the challenge.

7 Conclusion

The authors proposed a new global impostor selection method for a Deep belief Network (DBN) system in the multi-session i-vector speaker verification. The advantage of the proposed method is that only the background i-vector dataset is used in the selection process which make it suitable for using in the recent NIST i-vector challenge. The global selected impostors are further clustered and the cluster centroids are used as the final negative samples for discriminative DBN speaker models. The experimental results showed that the proposed impostor selection method increase the performance of the DBN system more than 7% in terms of minDCF and the final discriminative DBN models achieve a considerable performance in comparison to the conventional baseline system (more than 22% relative improvement).

Acknowledgement. This work has been partially funded by the Spanish Government projects TEC2010-21040-C02-01 and PCIN-2013-067.

References

1. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-End Factor Analysis for Speaker Verification. IEEE Transactions on Audio, Speech, and Language Processing 19, 788–798 (2011)
2. Prince, S., Elder, J.: Probabilistic Linear Discriminant Analysis for Inferences About Identity. In: IEEE 11th International Conference on Computer Vision, ICCV 2007 (2007)

3. Kenny, P.: Bayesian speaker verification with heavy tailed priors. In: IEEE Odyssey Speaker and Language Recognition Workshop (2010)
4. Brummer, N., Villiers, E.: The speaker partitioning problem. In: Proceedings of the Odyssey Speaker and Language Recognition Workshop, Brno, Czech Republic (2010)
5. Liu, G., Hasan, T., Boril, H., Hansen, J.: An investigation on back-end for speaker recognition in multi-session enrollment. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7755–7759 (2013)
6. Larcher, A., Lee, K., Ma, B., Li, H.: Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7673–7677 (2013)
7. Larcher, A., Bonastre, J.-F., Fauve, B., Lee, K., Lvy, C., Li, H., Mason, J., Parfait, J.-Y.: ALIZE 3.0 Open Source Toolkit for State-of-the-Art Speaker Recognition. In: Proc. Interspeech, pp. 2768–2771 (2013)
8. The 2013-2014 Speaker Recognition i-vector Machine Learning Challenge (2014)
9. Khoury, E., El Shafey, L., Ferras, M., Marcel, S.: Hierarchical speaker clustering methods for the NIST i-vector Challenge. In: Odyssey: The Speaker and Language Recognition Workshop, pp. 254–259 (2014)
10. Novoselov, S., Pekhovsky, T., Simonchik, K.: STC Speaker Recognition System for the NIST i-Vector Challenge. In: Odyssey: The Speaker and Language Recognition Workshop, pp. 231–240 (2014)
11. Vesnicer, B., Zganec-Gros, J., Dobrisek, S., Struc, V.: Incorporating Duration Information into I-Vector-Based Speaker-Recognition Systems. In: Odyssey: The Speaker and Language Recognition Workshop, pp. 241–248 (2014)
12. Khosravani, A., Homayounpour, M.: Linearly Constrained Minimum Variance for Robust I-vector Based Speaker Recognition. In: Odyssey: The Speaker and Language Recognition Workshop, pp. 249–253 (2014)
13. Ghahabi, O., Hernando, J.: i-Vector Modeling with Deep Belief Networks for Multi-Session Speaker Recognition. In: Odyssey: The Speaker and Language Recognition Workshop, pp. 305–310 (2014)
14. McLaren, M., Baker, B., Vogt, R., Sridharan, S.: Improved SVM speaker verification through data-driven background dataset collection. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009, pp. 4041–4044 (2009)
15. Liu, G., Suh, J.-W., Hansen, J.: A fast speaker verification with universal background support data selection. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4793–4796 (2012)
16. Ghahabi, O., Hernando, J.: Deep belief networks for i-vector based speaker recognition. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2014)
17. Hinton, G., Osindero, S., Teh, Y.-W.: A Fast Learning Algorithm for Deep Belief Nets. Neural Computation 18, 1527–1554 (2006)
18. Hinton, G., Salakhutdinov, R.: Reducing the Dimensionality of Data with Neural Networks. Science 313, 504–507 (2006)
19. Hinton, G.E.: A Practical Guide to Training Restricted Boltzmann Machines. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) NN: Tricks of the Trade, 2nd edn. LNCS, vol. 7700, pp. 599–619. Springer, Heidelberg (2012)
20. Dahl, G., Yu, D., Deng, L., Acero, A.: Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing 20, 30–42 (2012)